US009268454B2

(12) **United States Patent**
Hamilton, II et al.

(10) **Patent No.: US 9,268,454 B2**
(45) **Date of Patent: Feb. 23, 2016**

(54) **TRIGGER EVENT BASED DATA FEED OF VIRTUAL UNIVERSE DATA**

(75) Inventors: **Rick A. Hamilton, II**, Charlottesville, VA (US); **James R. Kozloski**, New Fairfield, CT (US); **Brian M. O'Connell**, Cary, NC (US); **Clifford A. Pickover**, Yorktown Heights, NY (US); **Keith R. Walker**, Austin, TX (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 644 days.

(21) Appl. No.: **12/120,629**

(22) Filed: **May 14, 2008**

(65) **Prior Publication Data**

US 2009/0288001 A1 Nov. 19, 2009

(51) **Int. Cl.**

G06F 3/0481	(2013.01)
G06F 17/30	(2006.01)
G07F 17/32	(2006.01)
H04L 29/06	(2006.01)
H04L 29/08	(2006.01)
A63F 13/30	(2014.01)
G06Q 30/02	(2012.01)

(52) **U.S. Cl.**

CPC ***G06F 3/0481*** (2013.01); ***G06F 17/3089*** (2013.01); *A63F 13/30* (2014.09); *A63F 2300/5553* (2013.01); *G06Q 30/02* (2013.01); *G07F 17/323* (2013.01); *H04L 29/06034* (2013.01); *H04L 29/0809* (2013.01)

(58) **Field of Classification Search**

CPC . G06F 3/0481; G06F 17/3089; G07F 17/323; G06Q 30/02; H04L 29/06034; H04L 29/0809; A63F 13/30; A63F 2300/5553

USPC 715/757–758; 463/42, 43

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,815,586	A	9/1998	Dobbins
6,020,885	A	2/2000	Honda
6,085,256	A	7/2000	Kitano et al.
6,281,898	B1	8/2001	Nikolovska et al.
6,396,509	B1 *	5/2002	Cheng 715/706

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO9509411	4/1995

OTHER PUBLICATIONS

Animal Crossing Wild World: Instruction Booklet for Nintendo DS, 2005, Nintendo.*

(Continued)

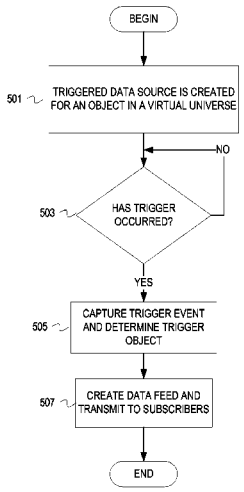
Primary Examiner — Eric J Bycer

(74) *Attorney, Agent, or Firm* — DeLizio Law, PLLC

(57) **ABSTRACT**

Business owners in a virtual universe may want to create a data source that transmits a data feed when certain parameters are met. Functionality can be implemented within a virtual universe to create a data feed when a trigger event occurs. The data feed may include data about the trigger event itself and/or about an avatar that caused the trigger. Triggered data feeds can be used to derive statistics, monitor use of objects and space and bill for such use, etc. In such cases, the user may be notified, such as via email, of the new data feed, as well as instructions on how to subscribe to it in their RSS reader or other client. Alternatively, users may be given individual feeds wherein additional feeds are distributed as separate categories of the user's feed as opposed to creating a separate feed.

4 Claims, 7 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

6,452,598	B1	9/2002	Rafey et al.	
6,518,989	B1	2/2003	Ishikawa	
6,704,784	B2	3/2004	Matsuda et al.	
7,036,082	B1 *	4/2006	Dalrymple et al.	715/757
7,266,616	B1	9/2007	Munshi et al.	
7,801,956	B1 *	9/2010	Cumberbatch et al.	709/204
2002/0113809	A1 *	8/2002	Akazawa et al.	345/706
2003/0005439	A1	1/2003	Rovira	
2003/0177187	A1 *	9/2003	Levine et al.	709/205
2003/0187749	A1 *	10/2003	Peled et al.	705/26
2004/0116183	A1	6/2004	Prindle	
2005/0179685	A1	8/2005	Kake et al.	
2005/0193341	A1	9/2005	Hayward et al.	
2006/0121991	A1 *	6/2006	Borinik et al.	463/43
2006/0129917	A1 *	6/2006	Volk et al.	715/513
2007/0113181	A1	5/2007	Blattner et al.	
2007/0192352	A1	8/2007	Levy	
2007/0192674	A1	8/2007	Bodin et al.	
2007/0209011	A1	9/2007	Padmanabhuni et al.	
2007/0220016	A1	9/2007	Estrada et al.	
2007/0225071	A1	9/2007	Van Luchene et al.	
2007/0255807	A1	11/2007	Hayashi et al.	
2007/0265090	A1 *	11/2007	Barsness et al.	463/42
2007/0265091	A1 *	11/2007	Aguilar et al.	463/42
2008/0194332	A1	8/2008	Kadikario et al.	
2008/0200253	A1 *	8/2008	Van Luchene et al.	463/42
2008/0207329	A1 *	8/2008	Wallace et al.	463/42
2008/0318678	A1 *	12/2008	Stivoric et al.	463/36
2009/0069084	A1 *	3/2009	Reece et al.	463/32
2009/0109213	A1	4/2009	Hamilton, II et al.	
2009/0124349	A1	5/2009	Dawson et al.	
2009/0144267	A1 *	6/2009	Cook et al.	707/5
2009/0158161	A1 *	6/2009	Gibbs et al.	715/733
2009/0164279	A1	6/2009	Black et al.	
2009/0209335	A1 *	8/2009	Pearce	463/30
2009/0215433	A1	8/2009	Cavanaugh et al.	
2009/0241039	A1 *	9/2009	Estevez et al.	715/757
2009/0287758	A1	11/2009	Hamilton, II et al.	
2010/0005424	A1 *	1/2010	Sundaresan et al.	715/849

OTHER PUBLICATIONS

"U.S. Appl. No. 12/120,625 Office Action", Dec. 28, 2009 , 13 pages.
"What is RSS?", *Software Garden* www.rss.softwaregarden.com/aboutrss.html Jul. 6, 2004 , 2 pages.
"U.S. Appl. No. 12/120,625 Final Office Action", Aug. 9, 2010 , 14 pages.
Dudeney, Gavin , "Streaming RSS Feeds in World", https://lists.secondlife.com/pipermail/educators/2006-September/002536.html (Date Obtained from internet Apr. 20, 2010) Sep. 24, 2006 , 3 Pages.
"U.S. Appl. No. 11/877,875 Final Office Action", Apr. 14, 2011 , 15 pages.
"U.S. Appl. No. 11/877,875 Office Action", Oct. 12, 2010 , 12 pages.
"U.S. Appl. No. 12/120,625 Office Action", Apr. 6, 2011 , 20 pages.

* cited by examiner

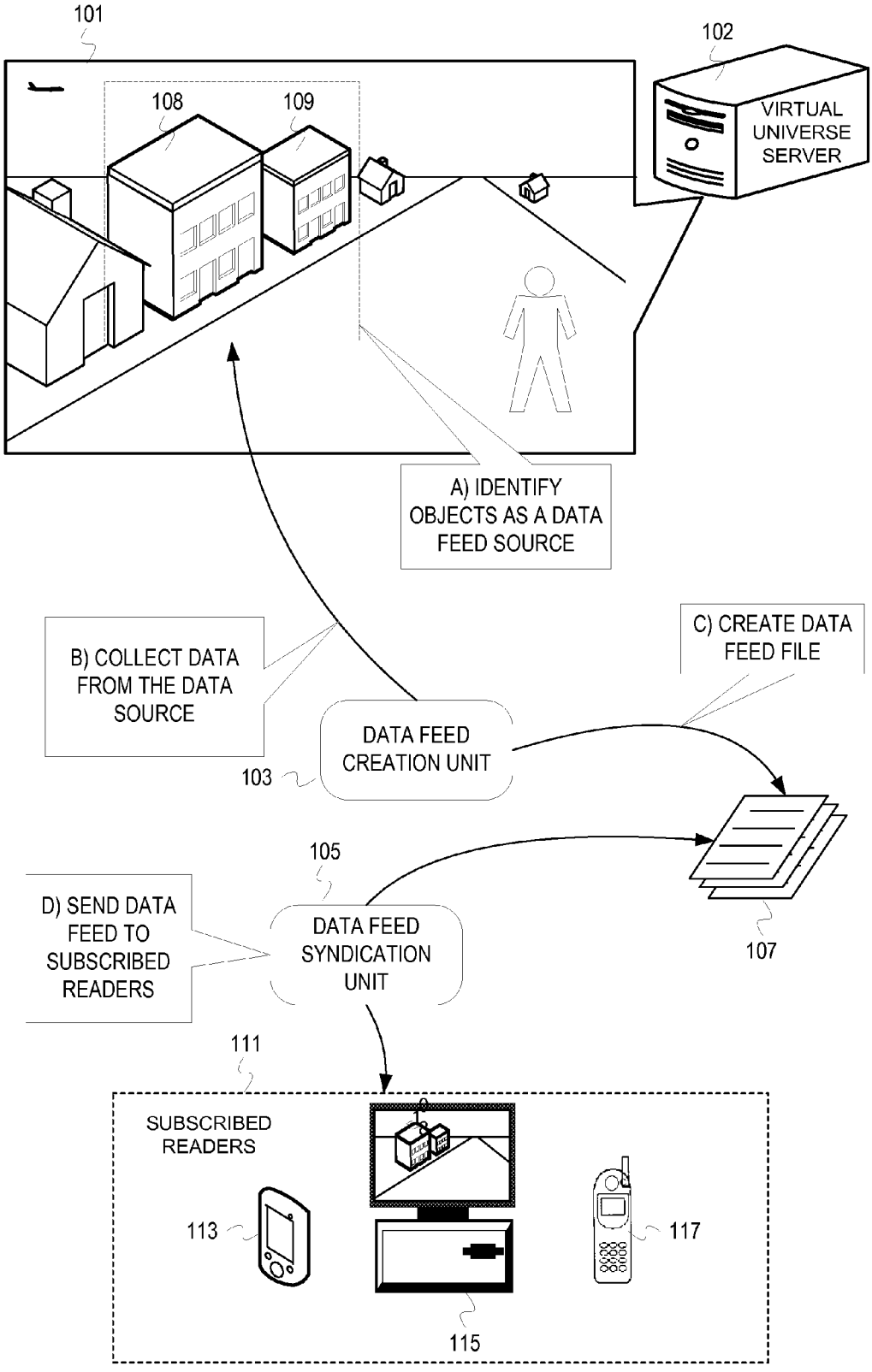

101
102
108
109
VIRTUAL UNIVERSE SERVER
A) IDENTIFY OBJECTS AS A DATA FEED SOURCE
B) COLLECT DATA FROM THE DATA SOURCE
C) CREATE DATA FEED FILE
DATA FEED CREATION UNIT
103
105
D) SEND DATA FEED TO SUBSCRIBED READERS
DATA FEED SYNDICATION UNIT
107
111
SUBSCRIBED READERS
113
115
117

FIG. 1

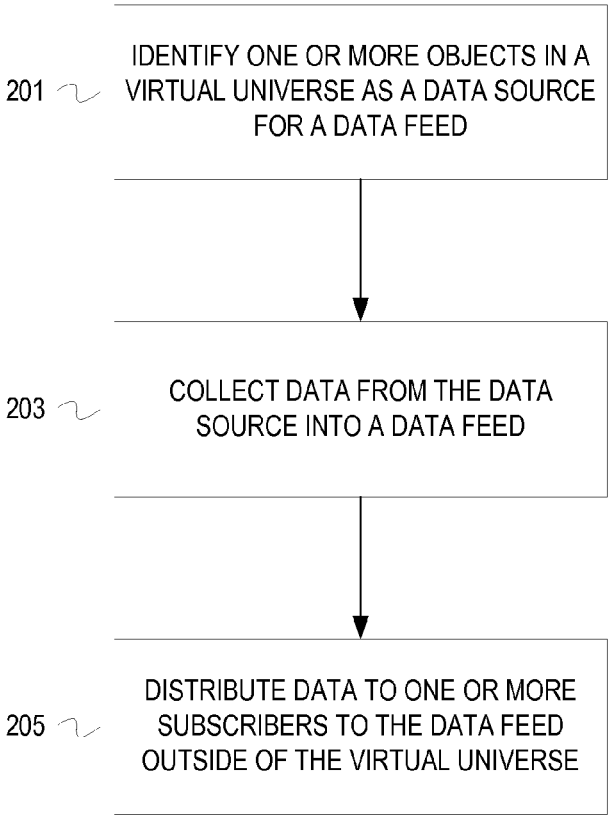

201
IDENTIFY ONE OR MORE OBJECTS IN A VIRTUAL UNIVERSE AS A DATA SOURCE FOR A DATA FEED
203
COLLECT DATA FROM THE DATA SOURCE INTO A DATA FEED
205
DISTRIBUTE DATA TO ONE OR MORE SUBSCRIBERS TO THE DATA FEED OUTSIDE OF THE VIRTUAL UNIVERSE

FIG. 2

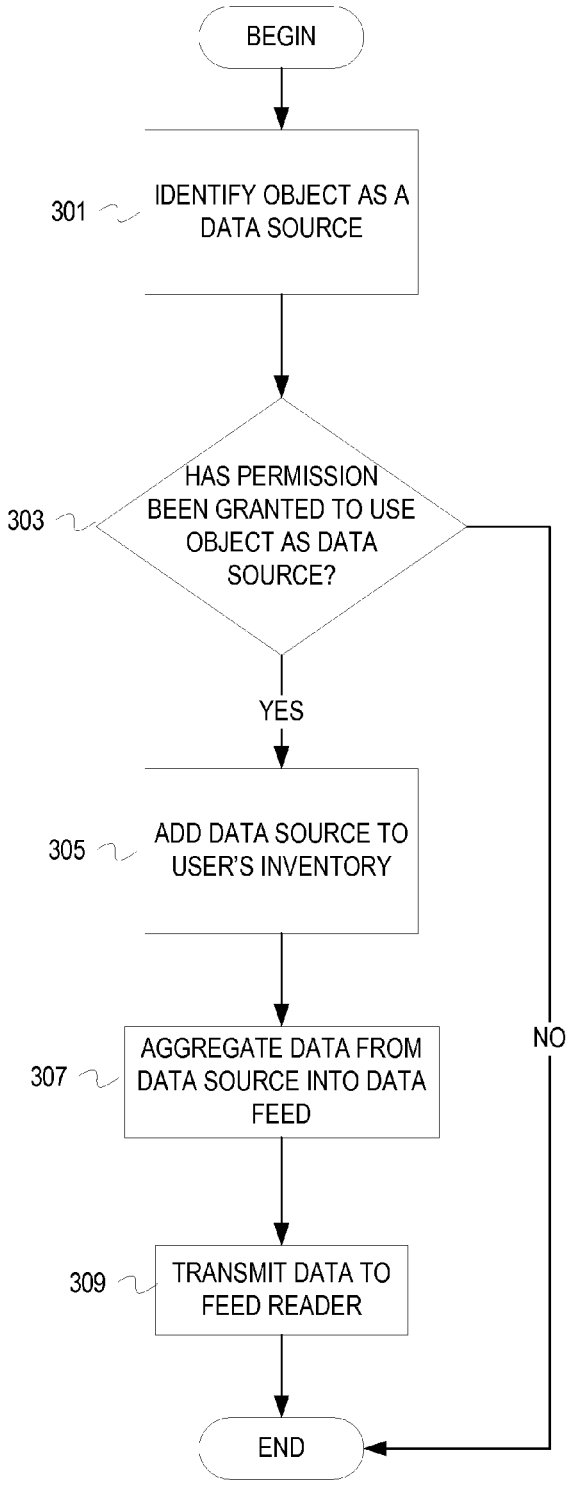

BEGIN
301
IDENTIFY OBJECT AS A DATA SOURCE
303
HAS PERMISSION BEEN GRANTED TO USE OBJECT AS DATA SOURCE?
YES
NO
305
ADD DATA SOURCE TO USER'S INVENTORY
307
AGGREGATE DATA FROM DATA SOURCE INTO DATA FEED
309
TRANSMIT DATA TO FEED READER
END

FIG. 3

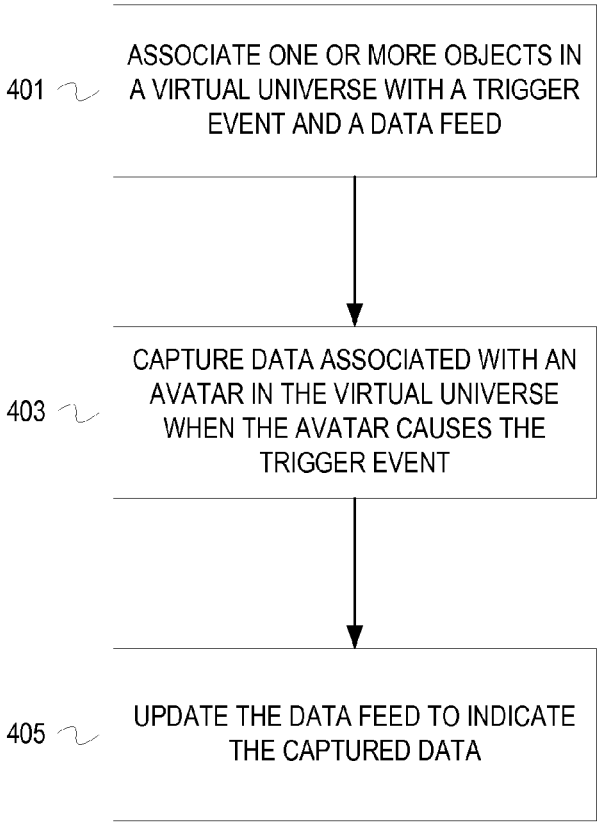

401
ASSOCIATE ONE OR MORE OBJECTS IN A VIRTUAL UNIVERSE WITH A TRIGGER EVENT AND A DATA FEED
403
CAPTURE DATA ASSOCIATED WITH AN AVATAR IN THE VIRTUAL UNIVERSE WHEN THE AVATAR CAUSES THE TRIGGER EVENT
405
UPDATE THE DATA FEED TO INDICATE THE CAPTURED DATA

FIG. 4

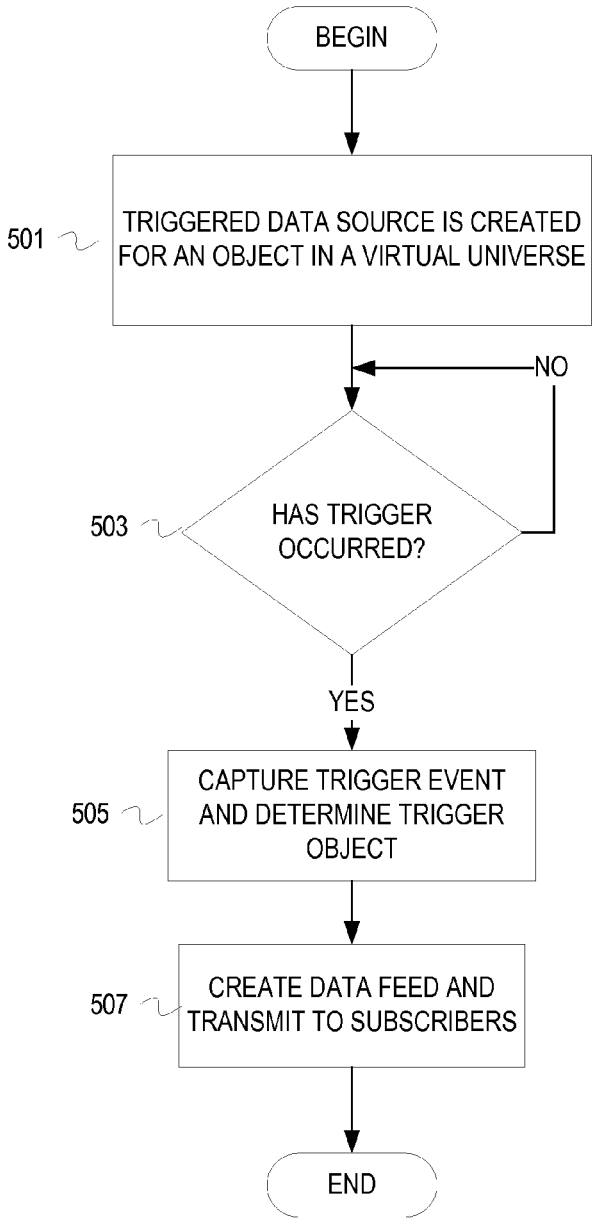

BEGIN
501
TRIGGERED DATA SOURCE IS CREATED FOR AN OBJECT IN A VIRTUAL UNIVERSE
NO
503
HAS TRIGGER OCCURRED?
YES
505
CAPTURE TRIGGER EVENT AND DETERMINE TRIGGER OBJECT
507
CREATE DATA FEED AND TRANSMIT TO SUBSCRIBERS
END

FIG. 5

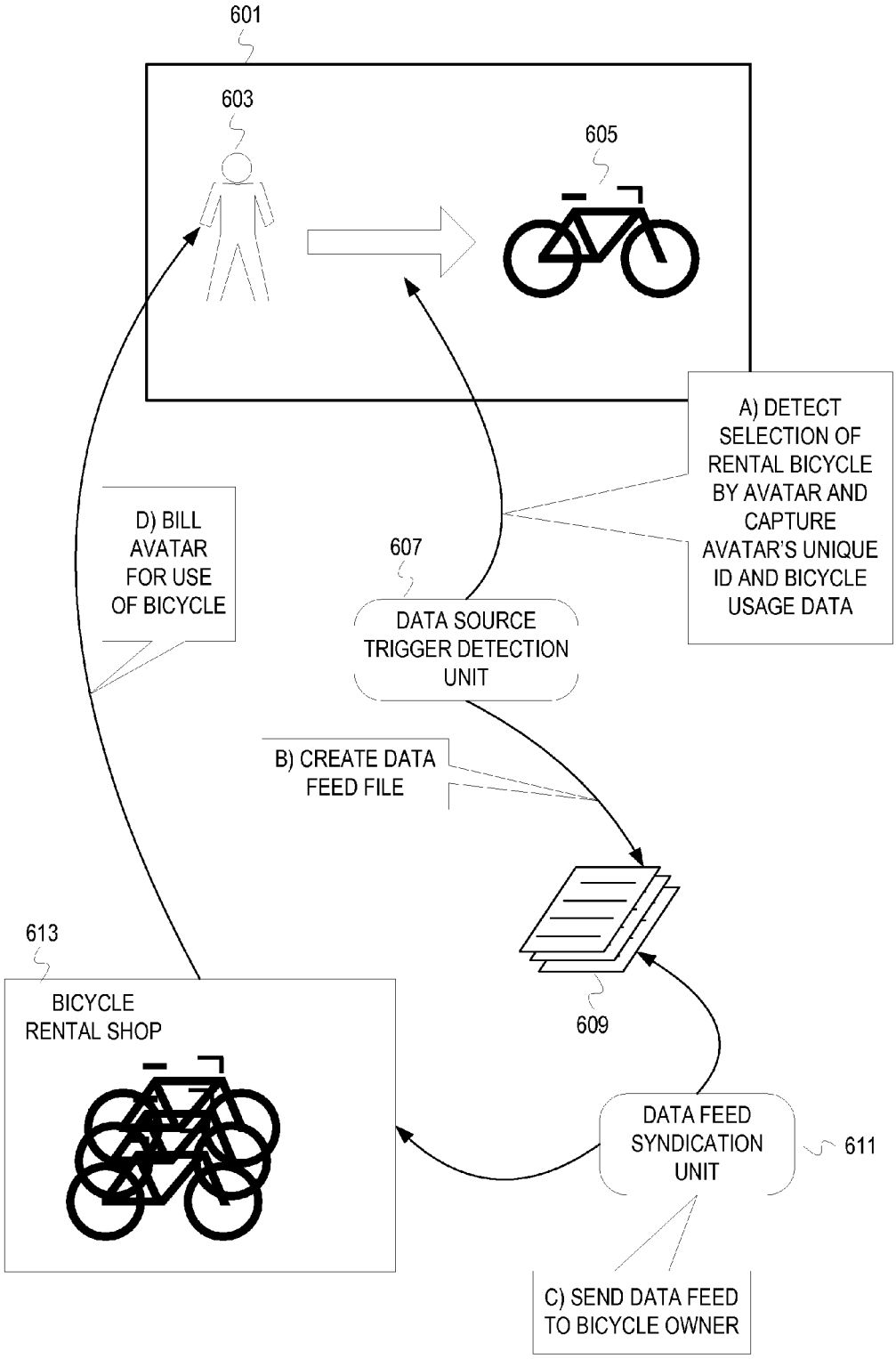

601
603
605
A) DETECT SELECTION OF RENTAL BICYCLE BY AVATAR AND CAPTURE AVATAR'S UNIQUE ID AND BICYCLE USAGE DATA
D) BILL AVATAR FOR USE OF BICYCLE
607
DATA SOURCE TRIGGER DETECTION UNIT
B) CREATE DATA FEED FILE
609
613
BICYCLE RENTAL SHOP
DATA FEED SYNDICATION UNIT
611
C) SEND DATA FEED TO BICYCLE OWNER

FIG. 6

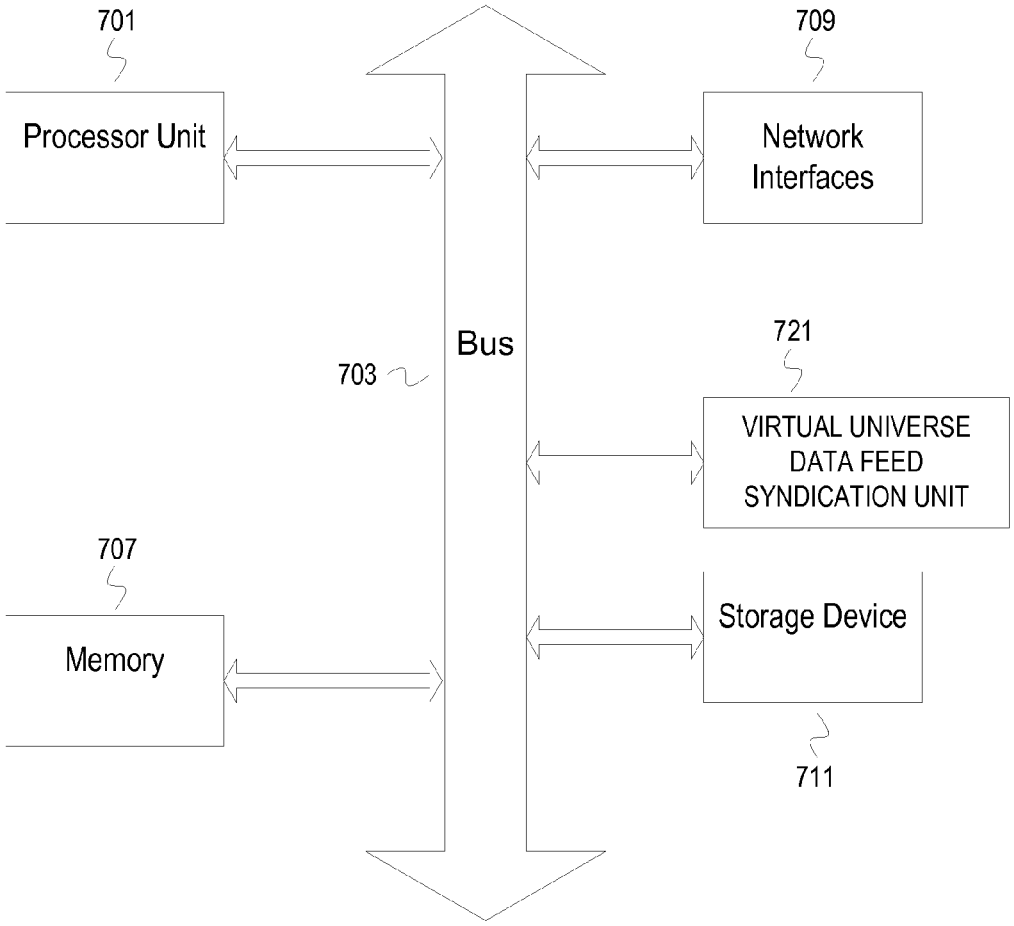

701
Processor Unit
709
Network Interfaces
Bus
703
721
VIRTUAL UNIVERSE DATA FEED SYNDICATION UNIT
707
Memory
Storage Device
711

FIG. 7

TRIGGER EVENT BASED DATA FEED OF VIRTUAL UNIVERSE DATA

BACKGROUND

Embodiments of the inventive subject matter generally relate to the field of virtual universe systems, and more particularly, to creating trigger events based web data feeds of virtual universe avatar data.

Virtual universe applications allow people to socialize and interact in a virtual universe. A virtual universe (VU) is a computer-based simulated environment intended for its residents to traverse, inhabit, and interact through the use of avatars. Many VUs are represented using 3-D graphics and landscapes, and are populated by many thousands of users, known as residents. Other terms for VUs include metaverses and "3D Internet."

Users of a virtual universe conduct business and engage in social interactions based on the representation of their avatars and the interaction of the avatars with asset, environment, and other avatar data served from the VU's simulated environment. Virtual economies are emerging as the rate of business and social interactions within VUs is rapidly increasing.

SUMMARY

Embodiments include a method directed to determining that a trigger event occurs in a virtual universe. The trigger event is associated with one or more virtual universe objects. Data associated with an avatar that causes that causes the trigger event in the virtual universe is captured. An update for a data feed utilizing the captured data associated with the avatar is generated. The trigger event is associated with the data feed.

BRIEF DESCRIPTION OF THE SEVERAL DRAWINGS

The present embodiments may be better understood, and numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

FIG. **1** depicts an example of data feed creation within a virtual universe **101**.

FIG. **2** depicts a flowchart of example operations for distributing a data feed to subscribers outside a VU.

FIG. **3** is a flowchart depicting example operations for creating and transmitting a data feed from a virtual universe.

FIG. **4** is a flowchart depicting example operations for creating a triggered data feed.

FIG. **5** is a flowchart depicting example operations for creating and distributing a triggered data feed.

FIG. **6** depicts an example use of triggered data feeds in a virtual universe.

FIG. **7** depicts an example computer system.

DESCRIPTION OF EMBODIMENT(S)

The description that follows includes exemplary systems, methods, techniques, instruction sequences and computer program products that embody techniques of the present inventive subject matter. However, it is understood that the described embodiments may be practiced without these specific details. For instance, although examples refer to Really Simple Syndication (RSS), embodiments can be implemented with any current or future technology similar to RSS, such as the Atom Syndication Format. In other instances, well-known instruction instances, protocols, structures and techniques have not been shown in detail in order not to obfuscate the description.

Business transactions are becoming more prevalent in virtual universes. Functionality can be implemented within a VU to allow syndication of data feeds to subscribers outside the VU. Feeds collect and serve data from one or more VU objects and locations based on changes made to the VU either by the VU itself, or by avatars in the VU. A location in a virtual universe can be an area defined by at least three coordinates, as an example. Data feeds from a VU can be used to create a VU or non-VU application with and/or using the data feeds, to derive data from the data feeds, to monitor use of objects or space and billing for such use based on the data feeds, etc.

FIG. **1** depicts an example of data feed creation within a virtual universe. A perspective **101** of a virtual universe is supported by a virtual universe server **102** and includes various graphical objects (e.g., buildings, avatars, vehicles, etc.) that make up the perspective **101**. In FIG. **1**, the perspective **101** comprises building graphical objects **108** and **109**. In stage A, the building **108** and the building **109** are identified as a source for a data feed ("data feed source"). An object is identified as a data feed source when a user or program selects the object and chooses an option to make the object a data feed source.

At stage B, a data feed creation unit **103**, collects data from the data feed source. Examples of collected data are graphical images, effects data, textures, coordinates, metadata, tracked usage, scripts, etc. As an example, data collected from buildings **108** and **109** may be statistical data (e.g., how many avatars passed by in a certain period of time, how many avatars entered one of the buildings, etc.), snapshots of avatars in the building, capture of the entire scene, etc.

At stage C, the data feed creation unit **103** creates a data feed **107** for syndication (e.g., using Really Simple Syndication (RSS)). As an example, the collected data may be formatted for syndication with data types separated into various fields (e.g., date and time formatted into a timestamp field, graphical images or metadata formatted into a description field, etc.).

At stage D, a data feed syndication unit **105** sends the data feed **107** to a set of subscribed readers **111**. Although a push type syndication model is described, embodiments can implement a pull type syndication model or a hybrid of push and pull syndication The subscribed readers **111** are applications running on various devices, examples of which include a computer **115**, a personal digital assistant (PDA) **113**, and a mobile phone **117**. A data feed reader running on computer **115** displays the data in the feed **107**. The feed **107** may not contain data about all objects shown in the perspective **101**. Five buildings are shown in the perspective **101**. The buildings **108** and **109** are selected as the data source. The feed **107** contains data pertaining to the buildings **108** and **109**, but not the other three buildings.

For example, the buildings **108** and **109** are listed for rent as storefronts by a property management company. The property management company identified the buildings **108** and **109** as a data feed to collect traffic data to show perspective tenants. The RSS feed **107** may include data such as how many avatars passed the buildings **108** and **109**, demographics of passersby, images of passersby, etc.

FIG. **2** depicts a flowchart of example operations for distributing a data feed to subscribers outside a VU. Flow begins at block **201**, where one or more objects in a VU are identified as a data source for a data feed. Examples of objects that can be used as data sources are buildings, avatars, vehicles, etc.

At block **203**, data from the data source is collected into a data feed. An object indicated as a data source may contain one or more additional objects. Data is collected for each additional object contained within the data source object. Examples of objects that contain one or more additional objects include buildings, store product displays, etc.

At block **205**, the data is distributed to one or more data feed subscribers outside of the virtual universe. In some embodiments, the data feed may be distributed at certain time intervals (e.g., daily). In other embodiments, the data feed may be a continuous stream of updates. In some embodiments, the data feed may be distributed to a subset of subscribers. The selection of these embodiments may be based on one or more conditions being met. For example, the default embodiment may be to distribute a data feed to all subscribers at an interval of 30 minutes, but a data feed containing a continuous stream of updates to a subset of subscribers (e.g., system administrators, authorities, object owners, etc.) may also be distributed.

FIG. **3** is a flowchart depicting example operations for creating and transmitting a data feed from a virtual universe. Flow begins at block **301**, where an object is identified as a data source. For example, a user selects an object and chooses an option to make the object a data source. As another example, a program selects an object based on a pattern match, such as a location visited by a threshold or percentage of avatars, an avatar engaging in sales above a threshold number or amount, etc. At block **303**, it is determined if permission has been granted to use the object as a data source. In one example, a request for permission to use the object as a data source can be sent to an owner of the object which may include samples of data that may be syndicated. In another example, the owner of the object can set a permission configuration associated with the object. If permission is granted to use the object as a data source, flow continues at block **305**. If permission is not granted to use the object as a data source, flow ends.

At block **305**, the data source is added to an inventory associated with the user's avatar. The inventory is a set of assets that are available to the user within the virtual universe. The inventory may contain tangible items such as clothing and vehicles and intangible items such as scripts and images. While the data source resides in the user's inventory, the user is considered to be subscribed to a data feed associated with the data source.

At block **307**, data from the data source is aggregated into a data feed.

At block **309**, the data is transmitted to subscribed feed readers outside the virtual universe and flow ends.

Data feed transmission can occur at certain intervals (i.e., daily). As an example, a user maintains an account in several VUs to work as a marketing consultant for several client companies that run virtual retail outlets in the VUs. In order to avoid logging into several VUs every day, the consultant implements an RSS feed from each of the VU retail centers. The RSS feeds provide periodic scene aggregates to an RSS reader application that allows switching between retail centers for daily site tours of each virtual store. The consultant may also create data feeds that collect statistical data over time, such as number and amount of sales, number of avatars in the store, demographics of avatars in the store, etc. A feed may be created to derive graphical data representing correlations between statistical data and avatar activity over time. For example, a feed can contain plots of avatar traffic flow in the store over a day. The data can be analyzed to determine the most and least popular areas of the store, navigation problems within the store, etc.

Data feed transmission can occur continuously. For example, a company is interested in introducing a certain number of new customers to a VU per day. The company wishes to allow new users to tour the VU, but due to server load cannot instantiate each guest avatar into the VU interface, or due to a lack of user willingness to install the VU client in order to tour the VU. Instead, the company can allow potential users to subscribe to an RSS feed. An RSS scene aggregator application provided by the company can render scenes at specific locations in each region of the VU. The web feed scene aggregator can also provide data sufficient for a limited view of each region. To limit computing resources to display the scene data, a partial representation of the scene data for each region may be collected into the data feed. For example, collected scene data may be limited to a certain number of objects and/or avatars, avatars may be represented by low resolution shapes, objects and/or avatars may be represented by outlines instead of graphics, etc.

In some cases, a user may want to create a data source that transmits a data feed when certain parameters are met. Functionality can be implemented within a virtual universe to create a data feed when a trigger event occurs. The data feed may include data about the trigger event itself and/or about an avatar that caused the trigger. Triggered data feeds can be used to derive statistics, monitor use of objects and space and bill for such use, etc. In such cases, the user may be notified, such as via email, of the new data feed, as well as instructions on how to subscribe to it in their RSS reader or other client. Alternatively, users may be given individual feeds wherein additional feeds are distributed as separate categories of the user's feed as opposed to creating a separate feed.

FIG. **4** is a flowchart depicting example operations for creating a triggered data feed. At block **401**, one or more objects in a virtual universe are associated with a trigger. A trigger event occurs when one or more indicated conditions associated with one or more objects have been met. Conditions are indicated by a user when the trigger event is created.

At block **403**, the trigger event for a data feed is indicated. The data feed is indicated by the user after creating the trigger event.

At block **405**, data associated with an avatar in the virtual universe that causes the trigger event is captured. Examples of data associated with an avatar include the avatar's name, the avatar's unique identifier, the email address of the avatar's owner, demographic data about the avatar and/or the user of the avatar, etc.

At block **407**, the data feed is updated to indicate the captured data. The data feed may be transmitted to a feed reader, used to derive other data, aggregated into another data feed, etc. In addition, the data feed may be updated to literally indicate the captured data, to indicate data derived from the captured data, to indicate a modified version of the captured data, etc.

FIG. **5** is a flowchart depicting example operations for creating and distributing a triggered data feed. Flow begins at block **501**, where a data feed trigger event is defined in a virtual universe. At block **502**, a data feed source is indicated for the data feed trigger event. For example, avatars, in general, are indicated as the data feed source for the trigger event. A data feed trigger event is defined by one or more conditions with one or more objects. The trigger event occurs when the one or more conditions have been satisfied. Example conditions include an avatar coming within a threshold distance from an object or location, an avatar picking up an object, an avatar entering a building, an avatar changing an attribute of an object, arrival of a particular time, etc. In addition, embodiments can implement nested dependent conditions. For

instance, a trigger event can be defined with a first condition that a current time be within a given range and a second condition that a given number of avatars be within a threshold distance of at least two of five objects identified as a data feed source.

At block **503**, it is determined if the trigger event has occurred. If the trigger event has occurred, flow continues at block **505**. If the trigger event has not occurred, flow continues at block **503**.

At block **505**, data about an avatar causing the trigger event and/or data about the trigger event is captured. An example of a trigger event associated with a concert venue is an avatar crossing a boundary that defines the area of the concert venue. The condition is satisfied and the trigger event occurs when the avatar walks through a doorway of the venue. Data captured when the avatar entered the venue may include a time stamp of when the trigger event occurs, a unique identifier of the avatar, etc. Another example of condition is an avatar touching an interactive display in a museum. The condition is satisfied and the trigger event occurs when the avatar touches the display. Data captured may include the amount of time the avatar interacted with the display, the number of times the display was touched throughout the day, etc.

In addition, a trigger event for a data feed may have more than one condition. For example, a store owner may want to analyze the effectiveness of a product display. Conditions can be set up to detect an avatar stopping in front of the display, touching the display, picking up a product from the display, etc.

At block **507**, the data feed is updated. Embodiments can accumulate data feed updates and update the data feed at certain intervals in batches, can update the data feed at every occurrence of a trigger event, etc. In addition, configurations can be set regarding the creation and transmission of data feeds for multiple objects and/or trigger events. For example, a user may wish to monitor trigger events for several different objects in a virtual universe. Each object may create an independent data feed. The user can set a configuration to receive the independent data feeds from each object or aggregate the independent feeds into one inclusive data feed.

FIG. **6** depicts an example use of triggered data feeds in a virtual universe. An avatar **603** and a rental bicycle **605** exist within a virtual universe **601**. The rental bicycle is an asset of a bicycle rental shop **613**. At stage A, selection of the rental bicycle **605** by avatar **603** is detected by a data source trigger detection unit **607**. The data source trigger detection unit **607** captures the unique identifier and bicycle usage data (e.g., model of bicycle chosen, how long the bicycle was used, distance traveled, a graphical map view of path traversed, etc.).

At stage B, a data feed file **609** is created by the data source trigger detection unit **607**. At stage C, a data feed syndication unit **611** sends the data feed to the bicycle rental shop **613**. In one example, the data feed is sent to an owner of the bicycle rental shop **613** via a feed reader outside the VU. In another example, the data feed is sent to a feed reader inside the VU that monitors use of the assets of the bicycle rental shop **613**. Notifications may be sent to an avatar belonging to the owner through communication services in the VU (e.g., instant message, email, etc.) At stage D, the bicycle rental shop **613** uses the data to bill the avatar for use of the bicycle. In some embodiments, the billing may be an automated response to the trigger event in the data feed. In other embodiments, the billing may be manually initiated by the owner of the asset. The bicycle rental shop can conduct market research by setting up a data feed to accumulate data over a period of time to analyze popularity of certain models of bicycles, characteristics of avatars that rent bicycles, distances the bicycles are ridden, paths taken, etc.

Triggered data feeds can be used to reward users for certain behaviors in a VU. As an example, an avatar may be given a small amount of revenue for entering a park and being exposed to an advertisement. As another example, an avatar may be given a free song download for attending a concert and being one of the first 100 avatars to enter the venue.

In addition, triggered data feeds can be used to monitor the use of spaces by avatars and charge owners of the avatars for use. A space is defined as an area within specific boundaries in the virtual universe. For example, an owner of a convention center in a virtual universe can define a space that is part of the convention center. The owner can monitor avatar activity within the space such as time spent in the space, corporate booths visited in the space, number of avatars in the space, etc.

It should be understood that the depicted flowchart are examples meant to aid in understanding embodiments and should not be used to limit embodiments or limit scope of the claims. Embodiments may perform additional operations, fewer operations, operations in a different order, operations in parallel, and some operations differently. For instance, referring to FIG. **3**, permission to use an object as a data source may not be implemented in certain embodiments because all objects within some VU may be used as a data source by default. In addition, with reference to FIG. **4**, a data feed may be conditioned upon multiple triggers. Additional operations may be performed to determine if additional trigger parameters have been met and/or a given time period has elapsed since the initial trigger.

Embodiments may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, embodiments of the inventive subject matter may take the form of a computer program product embodied in any tangible medium of expression having computer usable program code embodied in the medium. The described embodiments may be provided as a computer program product, or software, that may include a machine-readable medium having stored thereon instructions, which may be used to program a computer system (or other electronic device(s)) to perform a process according to embodiments, whether presently described or not, since every conceivable variation is not enumerated herein. A machine readable medium includes any mechanism for storing information in a form (e.g., software, processing application) readable by a machine (e.g., a computer). The machine-readable medium may include, but is not limited to, magnetic storage medium (e.g., floppy diskette); optical storage medium (e.g., CD-ROM); magneto-optical storage medium; read only memory (ROM); random access memory (RAM); erasable programmable memory (e.g., EPROM and EEPROM); flash memory; or other types of medium suitable for storing electronic instructions.

Computer program code for carrying out operations of the embodiments may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on a user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and

partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN), a personal area network (PAN), or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

FIG. **7** depicts an example computer system. A computer system includes a processor unit **701** (possibly including multiple processors, multiple cores, multiple nodes, and/or implementing multi-threading, etc.). The computer system includes memory **707**. The memory **707** may be system memory (e.g., one or more of cache, SRAM, DRAM, zero capacitor RAM, Twin Transistor RAM, eDRAM, EDO RAM, DDR RAM, EEPROM, NRAM, RRAM, SONOS, PRAM, etc.) or any one or more of the above already described possible realizations of machine-readable media. The computer system also includes a bus **703** (e.g., PCI, ISA, PCI-Express, HyperTransport®, InfiniBand®, NuBus, etc.), a network interface **709** (e.g., an ATM interface, an Ethernet interface, a Frame Relay interface, SONET interface, wireless interface, etc.), and a storage device(s) **711** (e.g., optical storage, magnetic storage, etc.). A virtual universe data feed syndication unit **721** performs the functionalities described herein. For instance, the virtual universe data feed syndication unit **721** can aggregate objects as a data source, collect data from the data source into a data feed, and transmit the data feed to subscribers. Any one of these functionalities may be partially (or entirely) implemented in hardware and/or on the processing unit **701**. For example, the functionality may be implemented with an application specific integrated circuit, in logic implemented in the processing unit **701**, in a co-processor on a peripheral device or card, etc. Further, realizations may include fewer or additional components not illustrated in FIG. **7** (e.g., video cards, audio cards, additional network interfaces, peripheral devices, etc.). The processor unit **701**, the storage device(s) **711**, and the network interface **709** are coupled to the bus **703**. Although illustrated as being coupled to the bus **703**, the memory **707** may be coupled to the processor unit **701**.

While the embodiments are described with reference to various implementations and exploitations, it will be understood that these embodiments are illustrative and that the scope of the inventive subject matter is not limited to them. In general, techniques as described herein may be implemented with facilities consistent with any hardware system or hardware systems. Many variations, modifications, additions, and improvements are possible.

Plural instances may be provided for components, operations or structures described herein as a single instance. Finally, boundaries between various components, operations and data stores are somewhat arbitrary, and particular operations are illustrated in the context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within the scope of the inventive subject matter. In general, structures and functionality presented as separate components in the exemplary configurations may be implemented as a combined structure or component. Similarly, structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements may fall within the scope of the inventive subject matter.

What is claimed is:

1. A method comprising:

responsive to determining that an owner of one or more virtual universe objects has granted permission to use the one or more virtual universe objects as a data source for a data feed:

determining that a trigger event occurs in a virtual universe, wherein the trigger event is associated with the one or more virtual universe objects, wherein the trigger event occurs after at least one condition has been satisfied, wherein the at least one condition comprises at least one of an avatar coming within a threshold distance of the one or more virtual universe objects, an avatar picking up the one or more virtual universe objects, and an avatar entering a building that is defined as the one or more virtual universe objects: and

responsive to the trigger event occurring in the virtual universe:

capturing data associated with an avatar that causes the trigger event in the virtual universe;

generating an update for the data feed utilizing the captured data associated with the avatar that causes the trigger event in the virtual universe, wherein the trigger event is associated with the data feed;

wherein the one or more virtual universe objects comprise a display of a product in the virtual universe;

wherein the at least one condition comprises at least one of detection of an avatar stopping in front of the display, detection of an avatar touching the display, and detection of an avatar picking up the product from the display; and

wherein the at least one condition comprises nested conditions that are dependent on each other, wherein all of the nested conditions are required to be satisfied for the trigger event to occur.

2. A method comprising:

requesting permission from an owner of one or more virtual universe objects to use the one or more virtual universe objects as a data feed source; and

responsive to receiving the permission from the owner of the one or more virtual universe objects to use the one or more virtual universe objects as a data feed source:

associating one or more conditions that involve the one or more virtual universe objects in a virtual universe with an indication of a data feed source, wherein the one or more conditions comprise at least one of an avatar coming within a threshold distance of the one or more virtual universe objects, an avatar picking up the one or more virtual universe objects, and an avatar entering a building that is defined as the one or more virtual universe objects;

collecting data from the data feed source in response to the one or more conditions being satisfied;

updating a data feed to indicate the collected data;

wherein the one or more conditions comprise nested conditions that are dependent on each other, wherein the collecting of the data for the data feed source is in response to all of the nested conditions being satisfied.

3. A computer program product for updating a data feed, the computer program product comprising:

one more computer readable tangible storage devices; and

program instructions, stored on at least one of the one or more storage devices,

responsive to determining that an owner of one or more virtual universe objects has granted permission to use

the one or more virtual universe objects as a data source for the data feed, to:

associate the one or more objects in a virtual universe with a trigger event, wherein the trigger event occurs after at least one condition has been satisfied, wherein the at least one condition comprises at least one of an avatar coming within a threshold distance of the one or more virtual universe objects, an avatar picking up the one or more virtual universe objects, and an avatar entering a building that is defined as the one or more virtual universe objects;

indicate the trigger event for the data feed;

responsive to the trigger event occurring in the virtual universe:

capture data associated with an avatar in the virtual universe that causes the trigger event;

update the data feed to indicate the captured data;

wherein the at least one condition comprises nested conditions that are dependent on each other, wherein all of the nested conditions are required to be satisfied for the trigger event to occur.

4. A system comprising:

one or more processors;

one or more computer-readable memories;

one or more computer readable tangible storage devices;

program instructions, stored on at least one of one or more storage devices for execution by at least one of the one or more processors via at least one of the one or more memories, to request permission from an owner of one or more virtual universe objects to use the one or more virtual universe objects as a source for a data feed; and

program instructions, stored on at least one of one or more storage devices for execution by at least one of the one or more processors via at least one of the one or more memories, responsive to receipt of the permission from the owner of the one or more virtual universe objects to use the one or more virtual universe objects as the source for the data feed, to:

associate the one or more objects in a virtual universe with a trigger events wherein the trigger event occurs after at least one condition has been satisfied, wherein the at least one condition comprises at least one of an avatar coming within a threshold distance of the one or more virtual universe objects, an avatar picking up the one or more virtual universe objects, and an avatar entering a building that is defined as the one or more virtual universe objects;

indicate the trigger event for the data feed;

responsive to the trigger event occurring in the virtual universe:

capture data associated with an avatar in the virtual universe that causes the trigger event;

update the data feed to indicate the captured data;

wherein the at least one condition comprises nested conditions that are dependent on each other, wherein all of the nested conditions are required to be satisfied for the trigger event to occur.

* * * * *